CS644 Final Project Report

CS644

**Zibin Guang**

**Minjuan Zhang**

# Structure of Oozie workflow

ERROR

Map Reduce  
On Schedule Airlines

Map Reduce  
Cancellation Reasons

Map Reduce  
Airports Tax Time

ERROR

ERROR

OK

OK

OK

# Map Reduce Design

## Map-Reduce: On Schedule Airlines

**Mapper:**

**Reading the data line by line inside the mapper function and less than 20 minutes delay, we will consider that flight is on time.**

**Reducer:**

**Read the data from mapper, and then calculate the probability for each airline to find out which airline delay the most or less.**

1. Mapper <key,value>:<UniqueCarrier,1 or 0>
2. The Mapper read the data line by line, ignore the first line and the NA data. If the data of the ArrDelay column which is less than or equal to 10 minutes, output:

<UniqueCarrier,1>, otherwise output: <UniqueCarrier,0>

1. Reducer <key,value>:<UniqueCarrier,probability> Probability = (# of 1) / (# of 1 and 0)
2. Reducer sum the values from the mapper of the same key, the sum will be the number of this airline when it is on schedule. And calculate the total number of 0 and 1, then calculate the on schedule probability of this airline.
3. Reducer then use the Comparator function do the sorting. After sorting, output the 3 airlines with the highest and lowest probability.
4. If the data is NULL, then output: There is no value can be used, so no output.

## Map-Reduce: Airports Taxi Time

**Mapper:**

**Read the data line by line, get the data from 20 and 21 columns for taxi in and taxi out.**

**Reducer:**

**Like on time schedule function, the function for taxi is normal on time/ average taxitime of each airport. Cleanup function will sort the list and output the 3 longest and shortest taxi time of the airport**

1. Mapper <key,value>: <IATA airport code, TaxiTime>: <Origin,TaxiOut> or <Dest,TaxiIn>
2. The Mapper read the data line by line, ignore the first line. If the data of the TaxiIn or the TaxiOut column is not NA, output: <IATA airport code, TaxiTime>
3. Reducer <key,value>: <IATA airport code, Average TaxiTime>
4. Reducer sum the value from the mapper of the same key (normal), and calculate the total times this key is found (all). Then do the equation: normal/all to calculath the average TaxiTime of each key.
5. Reducer then use the Comparator function do the sorting. After sorting, output the 3 airports with the longest and shortest average taxi time.
6. If the data is NULL, then output: There is no value can be used, so no output.

## Third Map-Reduce: Cancellation Reasons

**Mapper:**

**Read the data, get the data from column 23 which about the cancellation reasons.**

**Reducer:**

**Sum up all the reason by the keys which totally have 4. Clean up function helps us to sort the data and then output the most common reason**

1. Mapper <key,value>: < CancellationCode, 1>
2. The Mapper read the data line by line, ignore the first line. If the value of the Cancelled is 1 and the CancellationCode is not NA, output: < CancellationCode, 1>
3. Reducer <key,value>: < CancellationCode, sum of the 1s>
4. Reducer sum the value from the mapper of the same key.
5. Reducer then use the Comparator function do the sorting. After sorting, output the most common reason for flight cancellations.
6. If the data is NULL, then output: There is no the most common reason for flight cancellations.

# Increasing number of VMs

# 

Figure 2 shows that the number of VMs increase reduces the time. But until the certain point the communication interaction among VMs became time consuming so that the time on the computation won’t change any more.

# Increasing data size

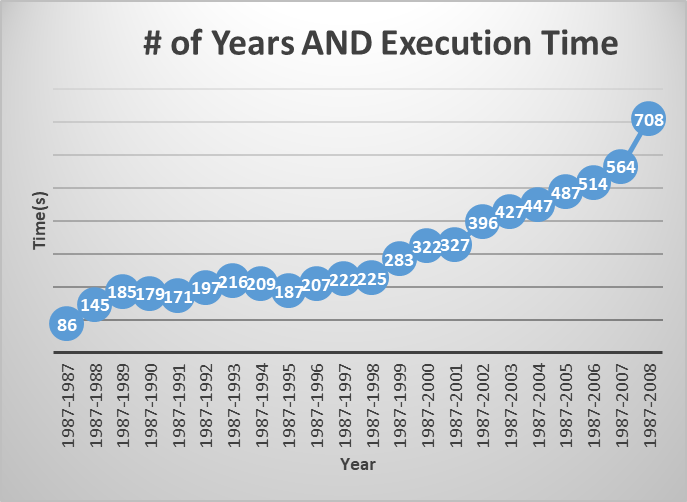
****

Figure 3 shows the later years more people choose to travel in flights illustrated by the slope of the increase curve so that the data volume increase and Oozie needs more time to calculate.

# Powerpoint – Hadoop Setup

<https://www.slideshare.net/MaggieZhang61/hadoop-presentation-135985781>

